

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Інститут телекомунікаційних систем

Кафедра Інформаційно-телекомунікаційних мереж

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Лариса ГЛОБА

« ____ » _____ 2020 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційно-комунікаційні
технології»
зі спеціальності 172 «Телекомунікації та радіотехніка»
на тему: «Метод побудови бізнес-процесів на базі онтології»**

Виконав:

студент VI курсу, групи ТІ-91мп

Бакай Андрій Миколайович _____

Керівник:

Професор кафедри ІТМ ІТС, професор, д.т.н.

Глоба Лариса Сергіївна _____

Рецензент:

Заступник директора з наукової роботи

НЦ «Мала академія наук України», д.т.н., с.н.с.,

Стрижак Олександр Євгенійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Інформаційно-телекомунікаційних мереж

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Лариса ГЛОБА

«__» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Бакаю Андрію Миколайовичу

1. Тема дисертації «Метод побудови бізнес-процесів на базі онотології», науковий керівник дисертації професор кафедри інформаційно-телекомунікаційних мереж ІТС Глоба Лариса Сергіївна, професор, д.т.н., затверджені наказом по університету від «03» листопада 2020 р. № 3208-с
2. Термін подання студентом дисертації 10.12.2020 р.
3. Вихідні дані до роботи: стандарти по автоматизації побудови робочих процесів, стандарт OWL
4. Зміст роботи:
 - 4.1. Провести аналіз методів побудови робочих процесів та визначити, що потрібно для автоматизації побудови робочих процесів
 - 4.2. Провести аналіз інструментальних засобів побудови робочих процесів
 - 4.3. Провести аналіз інструментальних засобів зберігання формалізованих описів в довгостроковому сховищі даних

4.4. Побудувати формалізований опис робочих процесів

4.5. Побудувати алгоритм формування робочих процесів на основі формалізованого опису

4.6. Провести експериментальне дослідження запропонованого методу формалізованого опису та алгоритму побудови робочих процесів за допомогою засобів сучасних СКБД та методів опису онтологічних моделей.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

1. Тема, актуальність, мета, задачі.

2. Аналіз інструментальних засобів побудови робочих процесів

3. Формалізований опис функціональних сервісів

4. Формалізований опис робочих процесів

5. Опис моделі та алгоритмів автоматизованої побудови робочих процесів

6. Загальні висновки

6. Дата видачі завдання 09 вересня 2019р

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Проведення аналізу методів побудови робочих процесів	11.09.2019-18.09.2019	виконано
2	Визначення актуальних питань, що потрібні для автоматизації побудови робочих процесів	19.09.2019-05.10.2019	виконано
3	Проведення аналізу інструментальних засобів побудови робочих процесів	06.10.2019-07.11.2019	виконано
4	Аналіз наявних методів формалізованого опису робочих процесів	08.11.2019-09.02.2020	виконано
5	Аналіз інструментальних засобів, за допомогою яких можна зберігати формалізовані описи в довгостроковому сховищі даних	10.02.2019-15.04.2020	виконано
6	Побудова формалізованого опису робочих процесів	16.04.2020-17.06.2020	виконано
7	Побудова алгоритму формування робочих процесів на основі формалізованого опису	18.06.2020-15.08.2020	виконано

8	Проведення експериментального дослідження запропонованих методів формалізованого опису та алгоритму побудови робочих процесів за допомогою засобів сучасних СКБД та методів опису онтологічних моделей	16.08.2020-19.10.2020	виконано
9	Проведення модельного експерименту	20.10.2020-09.12.2020	виконано

Студент

Андрій БАКАЙ

Науковий керівник дисертації

Лариса ГЛОБА

РЕФЕРАТ

Робота містить 70 сторінки, 10 рисунків. Було використано 54 джерела.

Цифрова економіка, яка є передумовою 4-ої індустріальної революції, вимагає стрімкого збільшення послуг, що надаються в цифровому глобальному просторі. Разом з тим, рівень автоматизації робочих процесів, які є основою різноманітних сервісів, не є досконалим, їх будують в недостатньо автоматизованому режимі через відсутність моделей та математичного апарату, який дозволив би легко виконувати певні операції взаємодії елементів таких моделей.

Мета роботи: підвищити ефективність побудови робочих процесів за рахунок автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків між функціональними сервісами на основі онтологічної моделі, яку легко корегувати, а також запропонованих інструментальних засобів побудови та модифікації бізнес процесів.

Об'єкт дослідження: процес побудови робочих процесів в сучасних інформаційних системах під час надання різноманітних сервісів кінцевим користувачам

Предмет дослідження: Математичні моделі та методи, засоби побудови робочих процесів, а також особливості проектування програмного забезпечення побудови бізнес-процесів.

Задачі:

Провести аналіз методів побудови робочих процесів та визначити, що потрібно для автоматизації побудови робочих процесів

Провести аналіз інструментальних засобів побудови робочих процесів

Побудувати алгоритм формування робочих процесів на основі формалізованого опису

Провести експериментальне дослідження запропонованого методу формалізованого опису та алгоритму побудови робочих процесів за допомогою засобів сучасних СКБД та методів опису онтологічних моделей.

Методи досліджень:

Основні методи дослідження загальної задачі – формалізація, системний підхід та експеримент. Методи формалізації використовуються для полегшення розробки комплексу програмного забезпечення. Методи системного підходу – для тестування виконання робочих процесів та їх компонентів. Експериментальні методи було використано для порівняння характеристик системи із іншими системами.

Теоретичний результат дослідження:

Проведено огляд та аналіз існуючих методів побудови робочих процесів, який дозволив зробити висновок, робочі процеси, які є основою різноманітних сервісів, будують в недостатньо автоматизованому режимі через відсутність моделей та математичного апарату, який дозволив би легко виконувати певні операції взаємодії елементів таких моделей.

Проведено аналіз інструментальних засобів, що дозволяють зберігати формалізовані описи в довгостроковому сховищі даних, що показав зручність використання нереляційних БД для задач даного типу.

Проаналізовано методи побудови робочих процесів на базі онтологічних моделей, які є недостатньо автоматизованими або зовсім неавтоматизовані.

Запропоновано метод формування робочих процесів на основі вхідних даних та за правилами відповідності параметрів, а також врахування областей допустимих значень та належності сервісів до одної предметної області, що дозволило підвищити ефективність побудови робочих процесів за рахунок

автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків

Практичний результат роботи:

Розроблено інструментальні засоби автоматизованої побудови робочих процесів із застосуванням мета-описів функціональних сервісів, які дозволяють зменшити кількість ручної праці користувачів.

Розроблено інструментальні засоби ведення реєстру функціональних сервісів, що дозволило легко зходити та модифікувати сервіси та робочі процеси, в яких вони використані.

Розроблено зручний для кінцевого користувача інтерфейс взаємодії з системою, який дозволяє легко формувати бізнес-процеси в термінах предметної області

Протестовано реалізацію запропонованого методу побудови робочих процесів. Результати експерименту показали доцільність використання даної системи у зв'язку зі швидкістю побудови робочих процесів та невимогливістю системи до обчислювальних ресурсів.

Застосування запропонованого підходу підтверджується прототипом програмного інструментарію, що забезпечує автоматизацію розробки робочих процесів, зосереджуючись на етапах планування та проектування, враховуючи як функціональні, так і нефункціональні вимоги, реалізуючи перетворення робочого процесу незалежно від обчислень в його модель виконання. Даний підхід дозволить генерувати бізнес процеси на основі заданих функціональних сервісів та обмежень накладених онтологією предметної області.

Публікації:

- Побудова робочих процесів на основі онтології предметної області

Ключові слова: онтологія, робочі процеси, функціональні сервіси, автоматизація процесу.

ABSTRACT

The work contains 70 pages, 10 figures. 54 sources have been used.

The digital economy, which is a prerequisite for the 4th industrial revolution, requires a rapid increase in the services provided in the digital global space. At the same time, the level of automation of work processes, which is the basis of various services, is not perfect, they are built in an insufficiently automated mode due to the lack of models and mathematical apparatus, which would allow it to be easy to perform certain interaction operations of elements of such models.

Goal: increase the efficiency of building workflows by automating their construction, reducing the amount of manual labor, and using a set of rules for building relationships between functional services in the form of an ontological model that is easy to adjust, as well as proposed tools for building and modifying business processes.

The application of the proposed approach is confirmed by a prototype software toolkit that provides automation of the development of work processes, focusing on the planning and design stages, considering both functional and non-functional requirements, implementing the transformation of the work process, regardless of the calculations in its implementation model. This approach will allow you to generate business processes based on the given functional services and restrictions imposed by the ontology of the subject area.

Object of research: the process of building workflows in modern information systems during the provision of various services to end users

Subject of research: Mathematical models and methods, tools for building workflows, as well as features of designing software for building business processes.

Tasks:

Analyze workflow construction methods and determine what is needed to automate workflow construction

Analyze the tools for building workflows

Build an algorithm for forming workflows based on a formalized description

Conduct an experimental study of the proposed method of formalized description and algorithm for building workflows using modern DBMS and methods of describing ontological models.

Research methods:

The main methods of research of the general problem are formalization, system approach and experiment. Formalization methods are used to facilitate the development of a set of software. Methods of system approach - for testing the performance of work processes and their components. Experimental methods were used to compare the characteristics of the system with other systems.

Publications:

- Construction of work processes based on the ontology of the subject area

Keywords: ontology, work processes, functional services, automation of the process.

ЗМІСТ

ВСТУП	13
РОЗДІЛ 1.....	15
Огляд робочих процесів та методів їх автоматизованої побудови	15
1.1. Огляд методів та інструментів аналізу робочих процесів.....	16
1.2. Засоби опису робочих процесів.....	19
1.3. Компонентами діаграми потоку даних є такі елементи:	21
Висновки	22
РОЗДІЛ 2.....	23
Онтологія	23
2.1. Поняття онтології.....	23
2.2. Елементи онтології	24
2.3. Мови онтологій	25
2.4. Засоби зберігання даних в цифровому світі.....	27
Висновки	31
РОЗДІЛ 3.....	32
Онтології робочих процесів та їх компонентів.....	32
3.1. Формалізований опис функціональних сервісів.....	32
3.2. Формалізований опис інфраструктури підтримки виконання робочих процесів	33
3.3. Автоматизована побудова робочих процесів із застосуванням онтології	34
3.4. Алгоритм формування робочого процесу	36
3.5. Аналіз онтології предметної області.....	36
3.6. Вибір користувачем сервісів при створенні нового робочого процесу	37
3.7. Операції алгебраїчної системи розрахунків	39
3.8. Режим відлагодження	43
Висновки	43
РОЗДІЛ 4.....	45

Тестування запропонованого підходу щодо побудови робочого процесу на прикладі підпису документу	45
4.1. Модель даних	48
Висновки	54
РОЗДІЛ 5.....	56
Стартап-проект	56
5.1. Опис ідеї проекту (товару, послуги, технології)	56
5.2. Технологічний аудит ідеї проекту	57
5.3. Аналіз ринкових можливостей запуску стартап-проекту	58
Висновки	62
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65

ПЕРЕЛІК СКОРОЧЕНЬ

БД	База даних
СКБД	Система керування базами даних
ДПД	Діаграма потоку даних
WfMS	Workflow management system

ВСТУП

Актуальність. Цифрова економіка, яка є передумовою 4-ої індустріальної революції, вимагає стрімкого збільшення послуг, що надаються в цифровому глобальному просторі. Разом з тим, рівень автоматизації робочих процесів, які є основою різноманітних сервісів, не є досконалим, їх будують в недостатньо автоматизованому режимі через відсутність моделей та математичного апарату, який дозволив би легко виконувати певні операції взаємодії елементів таких моделей.

Мета роботи: підвищити ефективність побудови робочих процесів за рахунок автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків між функціональними сервісами на основі онтологічної моделі, яку легко корегувати, а також запропонованих інструментальних засобів побудови та модифікації бізнес процесів.

Застосування запропонованого підходу підтверджується прототипом програмного інструментарію, що забезпечує автоматизацію розробки робочих процесів, зосереджуючись на етапах планування та проектування, враховуючи як функціональні, так і нефункціональні вимоги, реалізуючи перетворення робочого процесу незалежно від обчислень в його модель виконання. Даний підхід дозволить генерувати бізнес процеси на основі заданих функціональних сервісів та обмежень накладених онтологією предметної області.

Теоретичний результат дослідження:

Проведено огляд та аналіз існуючих методів побудови робочих процесів, який дозволив зробити висновок, робочі процеси, які є основою різноманітних сервісів, будують в недостатньо автоматизованому режимі через відсутність моделей та математичного апарату, який дозволив би легко виконувати певні операції взаємодії елементів таких моделей.

Проведено аналіз інструментальних засобів, що дозволяють зберігати формалізовані описи в довгостроковому сховищі даних, що показав зручність використання нереляційних БД для задач даного типу.

Проаналізовано методи побудови робочих процесів робочих процесів на базі онтологічних моделей, які є недостатньо автоматизованими або зовсім неавтоматизовані.

Запропоновано метод формування робочих процесів на основі вхідних даних та за правилами відповідності параметрів, а також врахування областей допустимих значень та належності сервісів до одної предметної області, що дозволило підвищити ефективність побудови робочих процесів за рахунок автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків

Практичний результат роботи:

Розроблено інструментальні засоби автоматизованої побудови робочих процесів із застосуванням мета-описів функціональних сервісів, які дозволяють зменшити кількість ручної праці користувачів.

Розроблено інструментальні засоби ведення реєстру функціональних сервісів, що дозволило легко зходити та модифікувати сервіси та робочі процеси, в яких вони використані.

Розроблено зручний для кінцевого користувача інтерфейс взаємодії з системою, який дозволяє легко формувати бізнес-процеси в термінах предметної області

Протестовано реалізацію запропонованого методу побудови робочих процесів. Результати експерименту показали доцільність використання даної системи у зв'язку зі швидкістю побудови робочих процесів та невимогливістю системи до обчислювальних ресурсів.

РОЗДІЛ 1.

ОГЛЯД РОБОЧИХ ПРОЦЕСІВ ТА МЕТОДІВ ЇХ АВТОМАТИЗОВАНОЇ ПОБУДОВИ

Дана робота є продовженням попередніх досліджень в напрямку автоматизованої побудови робочих процесів на базі онтологічних моделей [1].

Розвиток інформаційних технологій спричиняє трансформацію всіх сфер життєдіяльності суспільства. Сучасне суспільство є неможливим без цифрової економіки, формування якої є в даний час одним з пріоритетних напрямків. Для широкого загалу "цифрова економіка" буде означати новий рівень цифрових сервісів, коли в онлайн переходять оплати комунальних платежів, покупки і т.п., а для промисловості та бізнесу перехід в цифрову економіку отримав визначення в світі як Industry 4.0 - четверта індустріальна революція [2].

Сучасне бізнес середовище все більше надає значний обсяг послуг в цифровому просторі, тому телеком-оператори та провайдери онлайн-послуг мають гарантувати технологічну ефективність та інновації, постійно оновлюючи свої технології та послуги, адаптуючи та реконфігуруючи складні програмні системи, які функціонують в глобальному середовищі, надаючи послуги безперервно кінцевим користувачам.

Значну кількість робіт присвячено питанням автоматизації побудови робочих процесів, зокрема роботи 3-14.

Обчислювальні незалежні робочі процеси розробляють з використанням графічних стандартів, дозволяючи виконувати їх формалізацію з відображенням їх можливих потоків і переходів у схематичному вигляді. Аналіз показав, що на практиці обчислювальні незалежні робочі процеси зазвичай розробляються з використанням графічних позначень засобами, такими як BPMN 2.0 [5], UML AD (UML Діаграма Дій), USLD [6] та інструменти, такі як CA ERwin Process Modeler [7], Enterprise Architect [8] та MS Visio [9].

Недоліки BPMN 2.0 чітко описані в [15]. Центральний аргумент проти використання регулярного BPMN полягає в тому, що керування ресурсами може бути виражено лише через певні абстракції (актори, ролі тощо) або виконання завдань в діалозі.

Тим не менш, BPMN, забезпечуючи можливість обчислювальної незалежної від обчислювальних робочих процесів трансформації (Business Process Execution Language (BPEL)), широко поширена в промисловості.

1.1 Огляд методів та інструментів аналізу робочих процесів

Короткий огляд методів та інструментів аналізу робочих процесів показав, що існують два типи аналізу, які враховують обчислювальний процес:

1. Аналіз часу проектування (моделювання та перевірка). Можуть бути використані засоби моделювання Монте-Карло, а також аналіз мереж Петрі, оскільки існують підходи трансформації для BPMN [16], UML AD [17], EPC [18], BPEL [19] до мереж Петрі з їх подальшим аналізом. Діаграми USLD можуть бути проаналізовані за допомогою онтологічного аналізу послуг [20]

2. Аналіз часу виконання (наприклад, процес видобутку на основі журналів виконання) [21]

Для таких областей аналізу використовуються такі програмні засоби, як Pegasus, Cactus, ASKALON, GLUE і т.д. [22]. Всі згадані та проаналізовані поточні можливості для цього етапу завдання дуже обмежені. Недоліки методів та інструментів аналізу робочих процесів чітко описані в [23]. Центральною критикою є те, що етап аналізу вимог застосовується в основному вручну.

В роботі [12] розглянуто алгебру побудови робочих процесів, але це більш складний підхід, бо потребує детальної структуризації всіх даних та сервісів, встановлення повного переліку зв'язків, що утворює додаткову складність побудови алгебраїчної структури, особливо, коли система є територіально

розподіленою та має значну постійно змінювану кількість сервісів та робочих процесів.

В роботі [20] запропоновано інструментарій для побудови та аналізу робочих процесів, але все ж таки запропонований підхід не достатньо автоматизує саме формування послідовності обчислень в залежності від умов та обмежень предметної області.

Таким чином, можна зробити висновок, що стратегії адаптації систем під постійні зміни вимог користувачів за рахунок автоматизованого проектування та реінжинірингу послуг (тобто проектування та модифікацій існуючих бізнес-процесів) погано формалізовано та перевірено, вимагається декілька ітерацій за участі аналітиків, системних архітекторів, значних витрат часу та ресурсів.

Існує доволі багато інструментальних засобів для опису та побудови робочих процесів, але всі вони слабо автоматизовані та потребують участі людини. Рішень, які б дозволили виконувати реінжиніринг робочих процесів у зв'язку зі зміною мета-описів функціональних сервісів, на даний момент не існує.

Для побудови робочих процесів кластеризації цифрових даних в роботі [24] запропоновано застосування онтологічних моделей, що дозволило саме в залежності від характеристик потоку вхідних даних керувати процесом виконання обчислень. В даній сфері майже не використовують онтології для опису робочих процесів, хоча онтологія дає можливість формально описати процес, структурувати його частини та візуально відобразити його.

Враховуючи, що онтологічна модель робочих процесів дозволяє будувати останні, поєднуючи функціональні сервіси зв'язками у вигляді обмежень предметної області, які відображатимуть взаємодію між ними, а онтологічна модель функціональних сервісів зберігатиме їх формальні мета-описи, з'являється можливість утворення деякого інтелектуального середовища, на

основі якого можливо автоматично будувати робочі процеси в залежності від стану даних предметної області.

Перевагою онтологічного підходу є те, що онтологічна модель будується із застосуванням зв'язків у вигляді графів, дозволяючи застосовувати математичний апарат теорії графів для генерації робочих процесів в режимі реального часу, базуючись на описі вхідних та вихідних параметрів кожного функціонального сервісу.

Онтологія – опис концепцій та відношень в певній предметній області, придатний для автоматизованої обробки. Найчастіше онтологія спочатку описує базові елементи предметної області, а потім її розширюють до найменших деталей [25]. В онтології описуються всі зв'язки між цими елементами. Важливим фактором є те, що онтології створюють для вирішення певної задачі, тому її треба оцінювати з точки зору застосовності, ніж повноти [26].

Робочий процес (workflow) – схема організації послідовності виконання задач в межах єдиного сервісу, що надається кінцевим користувачам. Може включати пов'язані з ним обчислювальні підпроцеси, інформаційні залежності, послідовність прийняття рішень, проведення деяких обчислень.

Для відображення потоку робіт застосовують блок-схему або граф, який складається із операцій (функціональних сервісів), символів логіки, зв'язків. Послідовність виконання обчислень позначається стрілками.

Потік робіт з інформаційної точки зору – спосіб подання інформації до різних об'єктів, які приймають участь в робочому процесі [10].

Зазвичай робочий процес може бути описаний з використанням формальних або неформальних методів потокових діаграм, які показують спрямовані потоки між етапами обробки. Окремі кроки обробки або компоненти робочого процесу можуть бути в основному визначені трьома параметрами [27]:

- Опис входу: інформація, дані та ресурси, необхідні для робочого процесу.
- Правила перетворення: алгоритми, які можуть виконуватися людиною, програмою, чи обома.
- Опис виходу: інформація, дані та ресурси, оброблені робочим процесом та готові для передачі в наступні сервіси.

Компоненти можуть бути поєднані лише за умови, якщо вихідна інформація одного компоненту відповідає вхідній інформації наступного компоненту, а також задовольняє умовам предметної області. Таким чином, семантичний опис має включати обов'язково опис вхідних, вихідних даних та мета-опис функціональних сервісів, що виконуються в даному робочому процесі. Якщо існує кілька методів (функціональних сервісів) обробки одних і тих же даних, то потрібно додати мета-опис алгоритмів, які виконують задану функцію, а також бажано вказати характеристики ефективності виконання заданих функцій, такі як: точність, швидкість, та інші. [11]

Це пов'язано з тим, що інформація, якою оперує робочий процес, є слабо зв'язною різноструктурованою і може бути оброблена різними методами в залежності від характеристик потоку вхідних даних. Але з точки зору обробки даних, інформацію потрібно структурувати та навчити систему на конкретних даних для можливості більш ефективного використання різних методів.

1.2 Засоби опису робочих процесів

Робочий процес складається з організованої і повторюваної схеми діяльності, що забезпечується систематичною організацією ресурсів у процесах, що перетворюють дані, надають послуги або обробляють інформацію. Він може бути зображений як послідовність операцій, робота особи або групи, робота організації персоналу, або один або кілька простих або складних механізмів.

З більш абстрактної або вищої точки зору робочий процес може розглядатися як погляд або подання реальної роботи. Описуваний потік може посилатися на документ, послугу або продукт, який передається з одного кроку на інший.

Робочі процеси можна розглядати як один фундаментальний будівельний блок, який поєднується з іншими частинами структури організації, такими як інформаційні технології, команди, проекти та ієрархії.

Система управління робочими процесами (WfMS) - це програмна система для налаштування, виконання та моніторингу певної послідовності процесів і завдань, з цілями підвищення продуктивності, зниження витрат і покращення обміну інформацією в організації. Ці системи можуть бути орієнтовані на процеси або на дані, і вони можуть представляти робочий процес як графічні карти. Система управління робочим процесом може також включати розширюваний інтерфейс, щоб зовнішні програмні додатки могли бути інтегровані і забезпечувати підтримку робочих процесів широкої області, які забезпечують більш швидкий час відгуку та підвищення продуктивності.

Компоненти або підсистеми WfMS можна розділити на такі категорії:

- Система маршрутизації – це основна функція WfMS. Вона виконує маршрутизацію потоку інформації або документообігу, передає інформацію від одного робочого елемента до наступного. Ця система не обробляє дані.
- Система розподілу – ця функція є розширенням. Вона виявляє виняткові обставини і передає інформацію призначеним компонентам системи. Завдяки динамічному призначенню, робочий процес може призначати нові завдання недоробленим позиціям, щоб досягти продовження або балансу робочого навантаження в рамках робочого процесу.

- Система координації – ця функція координує одночасну діяльність, запобігаючи конфліктам ресурсів (виділення процесорного часу та оперативної пам'яті) або конфліктам пріоритетів.
- Система агентів – це компонент, що автоматично запускає сервіс, контролює його виконання та отримує від нього дані.
- Система помічників – цей компонент здійснює регулювання виборів наступних методів та надає пропозиції, який з методів може бути використаний наступним. Це основний метод, що використовується в штучному інтелекті.

Кожен робочий процес може бути описано діаграмою потоку даних.

Діаграма потоку даних (ДПД) – це спосіб представлення потоку даних процесу або системи (зазвичай це інформаційна система). ДПД також надає інформацію про результати та вхідні дані кожного суб'єкта та самого процесу. Діаграма потоку даних не має потоку керування, не існує правил прийняття рішень і не має петель. Конкретні операції на основі даних можуть бути представлені блок-схемою.

Для кожного потоку даних принаймні одна з кінцевих точок (джерела та / або призначення) повинна існувати в процесі. Уточнене представлення процесу може бути зроблено в іншій схемі потоку даних, яка поділяє цей процес на підпроцеси.

Схема потоку даних є частиною інструментів моделювання структурованого аналізу. При використанні UML діаграма діяльності зазвичай приймає роль діаграми потоку даних. Особливою формою плану потоку даних є план орієнтованого на сайт потоку даних.

1.3 Компонентами діаграми потоку даних є такі елементи:

- Процес (process) – частина системи, яка перетворює входи у виходи. Символом процесу є коло, овал, прямокутник або прямокутник із

закругленими кутами (відповідно до типу позначення). Процес називається одним словом, коротким реченням або фразою, яка явно виражає його сутність.

- Потік даних (data flow) – показує передачу інформації з однієї частини системи на іншу. Символом потоку є стрілка. Потік повинен мати ім'я, яке визначає, яка інформація (або який матеріал) переміщується. Винятки є потоками, де зрозуміло, яка інформація передається через суб'єкти, які пов'язані з цими потоками. Потік повинен передавати тільки один тип інформації. Стрілка показує напрямок потоку. Потоки зв'язують процеси, склади і термінатори.
- Склад даних (warehouse) – використовується для зберігання даних для подальшого використання. Символом сховища є дві горизонтальні лінії. Ім'я складу – це множинний іменник – він походить від вхідних і вихідних потоків складу. Склад не повинен бути просто файлом даних. Тому перегляд складу в ДПД не залежить від реалізації. Потік зі складу зазвичай являє собою читання даних, що зберігаються на складі, а потік на склад зазвичай виражає введення або оновлення даних (іноді також видалення даних). Склад представлений двома паралельними лініями, між якими розташовано назву пам'яті (вона може моделюватися як UML-буферний вузол).
- Термінал доступу (terminator) – це інтерфейс взаємодії системи із зовнішнім середовищем (людиною та/або іншою системою).

Висновки

В зв'язку з переходом до цифровизації в сучасному світі поширюється використання цифрових ресурсів для зберігання та оброблення користувацьких даних (та даних самих функціональних систем), для чого широко використовують робочі процеси та функціональні сервіси в різних сферах.

Проведений аналіз методів побудови та опису робочих процесів показує, що більшість з них будуються людиною в ручному або напів-автоматичному режимі, що дуже уповільнює процес створення самих робочих процесів.

Для відображення робочих процесів використовують теорію графів (кожна вершина – сервіс, а кожна стрілка – потік даних) та діаграм (набір елементів різних форм для відображення різних частин робочого процесу).

Для контролю виконання робочих процесів існують системи керування робочими процесами, які мають певні недоліки, зокрема відсутність пошуку функціональних сервісів в межах предметної області.

РОЗДІЛ 2. ОНТОЛОГІЯ

В інформатиці онтологія охоплює подання, формальне найменування та визначення категорій, властивостей та відносин між поняттями, даними та сутностями, які обґрунтовують один, багато або всі предметні області.

2.1. Поняття онтології

Кожний елемент онтології дозволяє обмежити складність, та організувати інформацію в дані та знання. Зі створенням нової онтології сподіваються покращити та полегшити вирішення проблем заданої предметної області.

Оскільки Google розпочав ініціативу під назвою Граф знань (Knowledge Graph [27]), значна кількість досліджень використовувала графів знань фрази як узагальнений термін. Незважаючи на відсутність чіткого визначення термінів графа знань, його іноді використовують як синонім онтології. Одна загальна інтерпретація полягає в тому, що граф знань є сукупністю взаємопов'язаних описів сутностей – об'єктів реального світу, подій, ситуацій або абстрактних понять. На відміну від онтологій, графи знань, такі як Google Knowledge Graph, часто містять великі обсяги фактичної інформації з менш формальною

семантикою. У деяких контекстах граф знань використовується для позначення будь-якої бази знань, яка представлена у вигляді графів.

Спільним для онтологій як в інформаційній науці, так і в філософії є спроба представити сутності, ідеї та події з усіма їх взаємозалежними властивостями і відносинами відповідно до системи категорій. В обох областях існує значна робота над проблемами інженерії онтології і дебати про те, наскільки можлива нормативна онтологія (наприклад, фундаменталізм і когерентність в філософії, BFO і Сус в штучному інтелекті). Прикладна онтологія вважається духовним спадкоємцем попередньої роботи у філософії, однак багато нинішніх зусиль більше стосуються детального формалізованого опису предметних областей.

Штучний інтелект зберігає найбільшу увагу стосовно прикладної онтології в підполях, таких як обробка природної мови в машинному перекладі та представлення знань, але редактори онтології часто використовуються в ряді областей, таких як освіта, без наміру внести свій внесок у AI.

2.2. Елементи онтології

Елементами онтології зазвичай є класи (поняття), екземпляри цих класів, їхні атрибути та значення цих властивостей.

Клас (поняття) – абстрактні групи, колекції або набори об'єктів. Можуть містити в собі екземпляри та/або інші класи.

Екземпляр – це основні, низько рівневі компоненти онтології.

Атрибути – елементи онтології, що мають ім'я і значення та використовуються для зберігання інформації, що є властивою для конкретного елемента. Значення атрибута може бути складеним типом даних.

Відношення – важливі компоненти онтології, так як вони показують як класи пов'язані між собою. Зазвичай відношенням є атрибут, у якого значенням є інший клас.

Функціональні терми – складні структури, сформовані з певних відносин, які можуть бути використані замість окремого терміну в твердженні

Обмеження – формально викладені описи того, що має бути істинним для того, щоб деяке твердження було прийнято в якості вхідних даних.

Правила – судження у вигляді пропозиції if-then (якщо-тоді), що описують логічні висновки, які можна зробити з твердження в певній формі.

Аксіоми – твердження (включаючи правила) у логічній формі, які разом складають загальну теорію, яку онтологія описує у своїй області застосування.

Події – зміна атрибутів або відносин.

2.3. Мови онтологій

Онтології зазвичай кодуються з використанням мов онтологій.

У комп'ютерних науках та штучному інтелекті мови онтологій є формальними мовами, що використовуються для побудови онтологій. Вони дозволяють кодувати знання про конкретні області і часто включають правила міркувань, які підтримують обробку цих знань. Мови онтологій, як правило, є декларативними, майже завжди узагальнення мов фреймів і зазвичай ґрунтуються на логіці першого порядку або на логіці опису.

Мова веб-онтологій (OWL) – це сім'я мов представлення знань для авторських онтологій. Онтології є формальним способом опису таксономій і мереж класифікації, які, по суті, визначають структуру знань для різних областей: іменники, що представляють класи об'єктів, і дієслова, що представляють відносини між об'єктами. Онтології нагадують ієрархії класів в об'єктно-орієнтованому програмуванні, але є кілька критичних відмінностей. Ієрархії класів призначені для представлення структур, що використовуються у вихідному коді, які розвиваються досить повільно (зазвичай щомісячні ревізії), тоді як онтології повинні представляти інформацію в Інтернеті і, як очікується, розвиватимуться майже постійно. Аналогічно, онтології, як правило, набагато

більш гнучкі, оскільки вони призначені для представлення інформації в Інтернеті, яка походить від різноманітних гетерогенних джерел даних. Ієрархії класів, з іншого боку, повинні бути досить статичними і спиратися на набагато менш різноманітні та структуровані джерела даних, такі як корпоративні бази даних.

Мови OWL характеризуються формальною семантикою. Вони побудовані на стандарті XML W3C (World Wide Web Consortium) для об'єктів, які називаються середовищем опису ресурсів (RDF). OWL та RDF залучили значний академічний, медичний та комерційний інтерес.

RDF спочатку розроблена як модель даних метаданих. Він став використовуватися в якості загального методу для концептуального опису або моделювання інформації, яка реалізована в веб-ресурсах, використовуючи різні синтаксичні позначення і формати серіалізації. Він також використовується в програмах управління знаннями.

Модель даних RDF схожа на класичні концептуальні підходи до моделювання (такі як діаграми об'єктних відносин або класів). Вона ґрунтується на ідеї створення тверджень про ресурси (зокрема, веб-ресурси) у виразах форми суб'єкт-предикат-об'єкт. Суб'єкт позначає ресурс, а предикат позначає ознаки або аспекти ресурсу і виражає зв'язок між суб'єктом і об'єктом.

Наприклад, один із способів представити поняття «Небо має колір синій» в RDF є як: суб'єкт, що позначає "небо", предикат, що позначає "має колір", і об'єкт, що позначає "синій". Тому RDF використовує суб'єкт замість об'єкта (або об'єктної сутності) на відміну від типового підходу моделі об'єктно-атрибутивного значення в об'єктно-орієнтованому дизайні: сутність (небо), атрибут (колір) і значення (синій).

RDF – абстрактна модель з кількома форматами серіалізації (тобто форматами файлів), тому конкретне кодування ресурсів або трійок варіюється від формату до формату.

Цей механізм опису ресурсів є основним компонентом семантичної активності W3C: еволюційний етап всесвітньої павутини, в якому автоматизоване програмне забезпечення може зберігати, обмінювати і використовувати інформацію, що може бути зчитана, оброблена та поширена по всьому інтернету машиною, у свою чергу, дозволяючи користувачам мати справу з інформацією з більшою ефективністю і впевненістю. Проста модель даних RDF і здатність моделювати розрізнені, абстрактні поняття також призвели до її все більшого використання в програмах управління знаннями, не пов'язаних із діяльністю Semantic Web.

2.4. Засоби зберігання даних в цифровому світі

В сучасному світі існує багато систем для збереження даних: від записів на папері до таких систем зберігання даних, як бази даних. Для систематизованого збереження великих обсягів даних дуже зручно використовувати бази даних, тому що вони дозволяють швидко знайти інформацію за заданими критеріями.

Бази даних (БД) – організований набір даних, що зберігається та може бути доступно в комп'ютерних системах. Якщо база даних більш складна, то вона розробляється з використанням формальних методів проектування та моделювання. В загальному випадку база даних має схеми, таблиці, подання, збережені процедури ті інші об'єкти. Сучасні бази даних крім самих даних зберігає ще і їх опис та засоби для обробки цих даних. Доступ до цих даних зазвичай забезпечується «системою керування базами даних».

Система керування базами даних (СКБД) – програмний додаток, що складається з інтегрованого набору комп'ютерних програм, які дозволяють користувачам взаємодіяти з однією або декількома базами даних і надає доступ

до всіх даних, що містяться в базі даних (хоча можуть існувати обмеження доступу до певних даних).

Програмне забезпечення СКБД додатково включає в себе основні засоби, що дозволяють адмініструвати БД. Сукупність СКБД, бази даних та додатків, що має зв'язок з базами даних може бути названо «системою баз даних». Часто термін «база даних» також використовується для визначення будь якої СКБД, системи баз даних чи додатків, пов'язаних з базою даних.

Існуючі СКБД забезпечують різні функції, які дозволяють керувати базою даних та її даними, які можна класифікувати за чотирма основними функціональними групами:

- Визначення даних – створення, модифікація та видалення визначень, що визначають організацію даних.
- Оновлення – вставка, модифікація та видалення фактичних даних.
- Отримання інформації – отримання інформації у формі, що безпосередньо використовується для подальшої обробки іншими додатками. Дані можуть бути доступними у формі, як вони зберігаються в базі даних або в новій формі, отриманій шляхом зміни або об'єднання існуючих даних з бази даних.
- Адміністрування – реєстрація та моніторинг користувачів, забезпечення безпеки даних, моніторинг продуктивності, підтримка цілісності даних, боротьба з контролем паралелізму та відновлення інформації, яка була пошкоджена деякою подією, наприклад, несподіваною відмовою системи.

І база даних, і її СКБД відповідають принципам конкретної моделі бази даних. «Система баз даних» посилається на модель бази даних, систему управління базами даних і базу даних.

Кожна система керування базами даних може бути класифікована в зв'язку з моделями баз даних, які вона підтримує. Реляційні бази даних стали

домінуючими в 1980-х роках. Ці дані моделюються у вигляді рядків та стовпців у ряді таблиць і в більшості випадків використовують SQL запити для зчитування та запису даних. В 2000-х набули популярності нереляційні бази даних, що називаються NoSQL, тому що вони використовують різні мови запитів.

NoSQL бази даних забезпечують механізм для зберігання і пошуку даних, які моделюються засобами, відмінними від табличних відносин, що використовуються в реляційних базах даних. Перевагами NoSQL баз даних є те, що вони легкі в проектуванні, їх легше масштабувати «горизонтально» для кластеризації, на відміну від реляційних БД. Структури даних, що використовуються базами даних NoSQL (наприклад, ключ-значення, широкий стовпець, графік або документ) відрізняються від тих, що використовуються за замовчуванням у реляційних базах даних – це робить деякі операції швидшими в NoSQL, на відміну від реляційних БД. Іноді структури даних, що використовуються базами даних NoSQL, також розглядають як «більш гнучкі», ніж таблиці реляційних баз даних. Бази даних NoSQL все частіше використовуються у великих даних і веб-додатках реального часу

Однією з NoSQL баз даних є MongoDB – крос-платформенна документо-орієнтована база даних. Серверний додаток використовує JSON-подібні документи зі схемою. База даних була розроблена корпорацією MongoDB Inc. та ліцензована під ліцензією Server Side Public License (SSPL).

Основними функціями та перевагами MongoDB є такі можливості:

- Спеціальні запити – MongoDB підтримує поля, пошуки за діапазоном значень і пошуки з використанням регулярних виразів. Запити можуть повертати конкретні поля документів, а також включати визначені користувачем функції JavaScript. Запити також можуть бути налаштовані для повернення випадкової вибірки результатів заданого розміру.

- Індексція – поля в документі MongoDB можуть бути індексовані за допомогою первинних і вторинних індексів.
- Реплікація – MongoDB забезпечує високу доступність даних за допомогою реплікації даних. Набір реплік складається з двох або більше копій даних. Кожен член набору реплік може діяти в ролі первинної або вторинної репліки в будь-який час. Всі запити читання та запису виконуються за замовчуванням на первинній репліці. Вторинні репліки підтримують копію даних первинного з використанням вбудованої реплікації. Коли первина репліка не доступна, інші репліки автоматично проводять «голосування», щоб визначити, яка вторинна репліка повинна стати первинною. За допомогою окремих налаштувань можна проводити операції зчитування із вторинних реплік для підвищення продуктивності.
- Балансування навантаження – MongoDB масштабується горизонтально, використовуючи шардінг. Користувач вибирає ключ шардінгу, який визначає, як будуть розділятися дані в колекції. MongoDB може працювати на декількох серверах, балансуючи навантаження або дублюючи дані, щоб підтримувати роботу системи в разі збою в роботі обладнання.
- Сховище даних – MongoDB може бути використана як файлова система, звана GridFS, з функціями балансування навантаження та реплікації даних на декількох машинах для зберігання файлів.
- Агрегація – MongoDB надає три способи виконання агрегації: консолідаційний конвеєр (the aggregation pipeline), функція map-reduce і методів агрегації для одного призначення (and single-purpose aggregation).
- Виконання серверного JavaScript коду – JavaScript може використовуватися в запитах, функціях агрегації (наприклад, MapReduce) і надсилатися безпосередньо до бази даних для виконання.

- Обмежені колекції – MongoDB підтримує колекції з фіксованими розмірами, які називаються обмеженими колекціями. Цей тип колекції підтримує порядок вставки і, як тільки зазначений розмір досягнуто, веде себе як кругова чергу.

Найбільш суттєвими перевагами системи баз даних MongoDB є те, що вона підтримує ієрархічну модель даних, яка має надає більш ефективний пошук та дозволяє гнучко модифікувати схеми зберігання документів.

Висновки

Аналіз існуючих засобів зберігання інформаційних ресурсів дозволяє зробити висновок, що онтології та методи організації інформації на їх основі є ефективними при стрімкому збільшенні інформаційних ресурсів.

Онтологічні моделі зберігаються в текстових форматах, що дозволяє їх зчитувати, обробляти та записувати за допомогою простого програмного коду.

Існують різні програмні засоби для створення, відображення та редагування онтології.

Проаналізувавши структури моделей збереження інформації в базах даних (реляційні та нереляційні), було визначено, що для збереження, систематизації та структуризації мета-описів найбільш ефективною буде MongoDB.

РОЗДІЛ 3.

ОНТОЛОГІЇ РОБОЧИХ ПРОЦЕСІВ ТА ЇХ КОМПОНЕНТІВ

В багатьох випадках одну й ту ж задачу можна вирішити різними методами. Характер обробки даних визначається імовірнісними властивостями спостережень, проведених на деяких даних, тому для опису всіх правил встановлення зв'язків на основі вхідних даних з метою визначення найбільш ефективного методу вирішення однієї і тієї ж задачі може бути використано онтологію.

3.1. Формалізований опис функціональних сервісів

Розглянемо запропонований підхід щодо автоматизації побудови робочих процесів на основі онтологій предметних областей для підвищення продуктивності побудови процесів за рахунок зменшення ручної праці та застосування математичних моделей.

Функціональний сервіс (S) – сервіс, що приймає на вхід набір даних, виконує операцію над ними та віддає на вихід нові дані. Збереження функціональних сервісів виконується за допомогою файлової системи, а мета-описи сервісів знаходяться в базі даних. Всі мета-описи функціональних сервісів кожної предметної області зберігаються в окремих колекціях бази даних. Сервіси можуть бути простими та складними.

Розглянемо модель простого сервісу: $S = \{ Code, M, P \}$, де:

$Code$ – програмний код функціонального сервісу

M – мета-опис функціонального сервісу

P – фізичний шлях до функціонального сервісу

Мета-опис функціонального сервісу (M) – це опис функції, її вхідних та вихідних параметрів для подальшого відображення в користувацькому інтерфейсі та для врахування обмежень при побудові зв'язків у робочому процесі.

Мета-опис простого сервісу може бути представлено у вигляді:

$M = \{ ID, F, \{I_1, \dots, I_n\}, \{O_1, \dots, O_n\}, T \}$, де:

$M = \{ ID, F, \{I_1, \dots, I_n\}, \{O_1, \dots, O_n\}, T \}$ – мета-опис, що включає такі параметри:

ID – ідентифікатор мета-опису, наприклад:
`ObjectID("5cdd4fd0598ec622d023e048")`

$F = \{ N, D, K, P \}$ – множина параметрів для опису функціонального сервісу;

$I_i = \{ N, D, K, T \}$ – множина параметрів для опису вхідних параметрів;

$O_i = \{ N, D, K, T \}$ – множина параметрів для опису вихідних параметрів, де:

N – назва об'єкту (функціонального сервісу, вхідного/вихідного параметрів), наприклад «*checkBankAccount*»

D – опис об'єкту; характеристика обчислювальних операцій функціонального сервісу, наприклад: «Сервіс перевіряє, чи існує дана картка в банку призначення»

K – зручні для розуміння людини ключові слова для швидкого пошуку сервісу чи параметрів

P – шлях до функціонального сервісу «./bank/checkBankAccount.js»

T – тип сервісу: «function», «process»; тип змінної «number», «string» ...

$Proc = \{ CPU, R, t_{max}, \dots \}$ – опис нефункціональних параметрів сервісу, де:

C – кількість обчислювальних ядер, що потрібні для виконання сервісу

R – кількість оперативної пам'яті

t_{max} – максимальний час для виконання сервісу

... – інші параметри

3.2. Формалізований опис інфраструктури підтримки виконання робочих процесів

В автоматизованих системах виконання робочих процесів виконується за допомогою систем керування робочими процесами та можуть бути описані за допомогою діаграм потоку даних.

Формально система керування робочими процесами може бути предсталена у вигляді:

$$WfMS = \{ S_r, S_d, S_c, S_a, S_{ai} \}, \text{ де:}$$

S_r – система маршрутизації, виконує маршрутизацію потоку інформації або документообігу, передає інформацію від одного робочого елемента до наступного;

S_d – система розподілу, виявляє виняткові обставини і передає інформацію призначеним компонентам системи;

S_c – система координації, координує одночасну діяльність, запобігаючи конфліктам ресурсів (виділення процесорного часу та оперативної пам'яті) або конфліктам пріоритетів;

S_a – система агентів, автоматично запускає сервіс, контролює його виконання та отримує від нього дані;

S_{ai} – система помічників, здійснює регулювання виборів наступних методів та надає пропозиції, який з методів може бути використаний наступним.

Компоненти діаграми потоку даних може бути записано у вигляді:

$$DFD = \{ D_p, D_{df}, D_w, D_t \}, \text{ де:}$$

D_p – процес, частина системи, яка перетворює входи у виходи;

D_{df} – потік даних, показує передачу інформації з однієї системи на іншу;

D_w – склад даних, використовується для зберігання даних для подальшого використання

D_t – термінал доступу, інтерфейс взаємодії системи із зовнішнім середовищем.

3.3. Автоматизована побудова робочих процесів із застосуванням онтології

Для побудови робочого процесу потрібно виконати операцію перетину кожний з кожним для всіх сервісів за умови існування логічного зв'язку, а також

зв'язку по даним між ними. При цьому вхідні дані визначатимуть той сервіс, з якого починаються обчислення за умови співпадіння даних з їх мета-описами в сервісі.

Формально побудова робочого процесу може бути представлено у вигляді:

$$P = \{ \forall M_{вик,i} \cap \forall M_{вик,j} / \exists L_{ij} \}, \text{ де:}$$

$P = \{ \forall M_{вик,i} \cap \forall M_{вик,j} / L_{ij} \}$ – згенерований робочий процес, що складається

з:

$M_{вик} = \{ M_{зад} \cap M_{обл} \}$ – вибір сервісів, що використані в робочому процесі,

де:

$M_{вик}$ – функціональні сервіси, що використані в робочому процесі

$M_{зад}$ – задані користувачем функціональні сервіси

$M_{обл}$ – функціональні сервіси даної предметної області

$L_{ij} = \{ M_{вик,i}.O = M_{вик,j}.I / VM_{вик,i}.O \in VM_{вик,j}.I \}$ – визначення зв'язків робочого

процесу за правилом сумісних вхідних та вихідних параметрів, де:

L_{ij} – зв'язки робочого процесу

$M_{вик,i}.O$ – вихідний параметр і-того функціонального сервісу

$M_{вик,j}.I$ – вхідний параметр j-того функціонального сервісу

$VM_{вик,i}.O$ – допустимі значення вихідного параметру і-того сервісу

$VM_{вик,j}.I$ – допустимі значення вхідного параметру j-того сервісу

$M_{обл} = \{ M_{всі} \in SA \}$ – вибір всіх сервісів, що належать предметній області,

де:

$M_{обл}$ – функціональні сервіси заданої предметної області

$M_{всі}$ – всі, доступні в репозиторії, функціональні сервіси

SA – задана предметна область

3.4. Алгоритм формування робочого процесу

Кожен робочий процес може бути створено спеціалістом предметної області за вибраними параметрами функціональних сервісів або вибрано серед запропонованих системою автоматично згенерованих варіантів.

Алгоритм формування робочих процесів включає в себе такі етапи:

1. Аналіз онтології предметної області та функціональних сервісів системою та зберігання цих даних до бази даних
2. Вибір користувачем сервісів, що будуть використані в робочому процесі, серед доступних сервісів в базі знань за допомогою веб-інтерфейсу
3. Формування системою сценарію виконання робочого процесу на основі набору сервісів та правил поєднання цих сервісів, заданих спеціалістом чи предметною областю та зберігання робочого процесу в базу даних

3.5. Аналіз онтології предметної області

Опис предметної області може бути задано у зручному для спеціаліста вигляді за допомогою онтології.

При імпортуванні онтології системою буде проаналізовано елементи онтології та зв'язки між ними. Після цього, зібрана інформація буде передана до бази знань.



Рис 3.1. Етапи побудови робочого процесу

3.6. Вибір користувачем сервісів при створенні нового робочого процесу

При вході до веб-інтерфейсу користувачу буде запропоновано вибрати поточний бізнес-процес чи створити новий.

При створенні нового робочого процесу користувачу буде запропоновано вибрати сервіси з репозиторію сервісів заданої предметної області, враховуючи знання з бази знань.

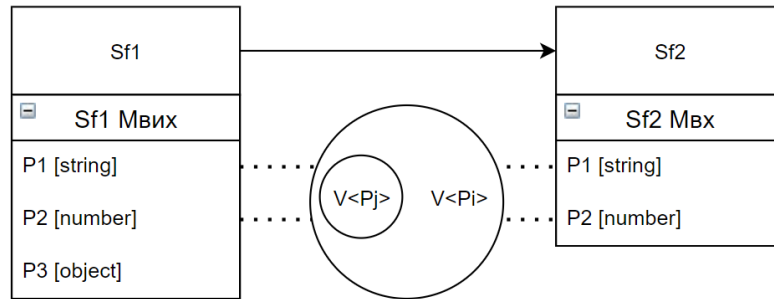


Рис 3.2. Умови вибору сервісів

Необхідними умовами для створення потоку даних між функціональними сервісами є:

- Обидва функціональні сервіси належать до однієї предметної області або один з функціональних сервісів є універсальним сервісом та доступний для використання в будь-якій предметній області
- Тип вихідного параметру першого функціонального сервісу дорівнює типу вхідного параметру другого функціонального сервісу. Може бути представлено у вигляді

$$L_{Pij} = \forall S_{f2}P_i \{ \exists S_{f1}P_j \mid T_{<S_{f2}P_i>} = T_{<S_{f1}P_j>} \}, \text{ де:}$$

L_{Pij} – встановлений зв'язок передачі i -того параметру першого сервісу в j -тий параметр другого сервісу

$S_{f2}P_i$ – i -тий параметр наступного функціонального сервісу

$T_{<S_{f2}P_i>}$ – тип i -того параметру наступного функціонального сервісу

$T_{<S_{f1}P_j>}$ – тип j -того параметру попереднього сервісу

- Область допустимих значень вихідного параметру першого сервісу належить області допустимих значень вхідного параметру другого сервісу

$$\forall L_{Pij} \{ V_{<S_{f1}P_j>} \in V_{<S_{f2}P_i>} \}, \text{ де:}$$

$V_{\langle Sf1Pj \rangle}$ – допустимі значення вихідного параметру першого сервісу

$V_{\langle Sf2Pi \rangle}$ – допустимі значення вхідного параметру другого сервісу

Достатніми умовами для створення потоку даних між функціональними сервісами є:

- Наявність такого зв'язку між заданими функціональними сервісами в інших робочих процесах
- Наявність такого зв'язку в списку наперед визначених спеціалістом зв'язків даної предметної області

Для більш швидкого пошуку потрібного компоненту, кожен сервіс має опис та ключові слова у зрозумілому для людини вигляді.

3.7. Операції алгебраїчної системи розрахунків

Під час формування сценарії робочого процесу користувачем може бути вибрано такі операції поєднання сервісів

Операція послідовного виконання

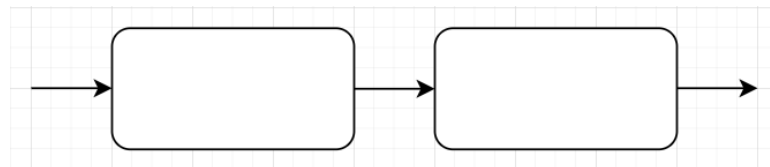


Рис 3.3. Операція послідовного виконання

Операція послідовного виконання - E_{con} , це операція, що описує процес передачі вихідних даних одного функціонального сервісу на вхід іншого функціонального сервісу.

Тоді математична модель даної операції буде виглядати так

$$S_f[E_{con}(S_{f1}, S_{f2})] = \{ M_{Bx} = S_{f1}M_{Bx}, M_{Bix} = S_{f2}M_{Bix} \}, \text{ де:}$$

M_{Bx} – вхідні дані блоку, що буде отримано

M_{Bix} – вихідні дані блоку, що буде отримано

$S_{f1}M_{вх}$ – вхідні дані першого функціонального сервісу

$S_{f2}M_{вих}$ – вихідні дані другого функціонального сервісу

Операція паралельного виконання

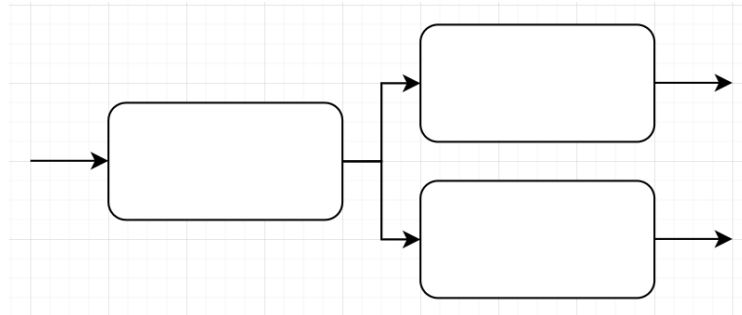


Рис 3.4. Операція паралельного виконання

Операція паралельного виконання - E_{par} , це операція, що описує процес передачі вихідних даних одного функціонального сервісу на вхід кількох інших функціональних сервісів.

Тоді математична модель даної операції буде виглядати так

$$S_f[E_{par}(S_{f1}, \{ S_{fi} \})] = \{ M_{вх} = S_{f1}M_{вх}, M_{вих} = \{ S_{fi}M_{вих} \} \}, \text{ де:}$$

$M_{вх}$ – вхідні дані блоку, що буде отримано

$M_{вих}$ – вихідні дані блоку, що буде отримано

$S_{f1}M_{вх}$ – вхідні дані першого функціонального сервісу

$S_{fi}M_{вих}$ – вихідні дані інших функціональних сервісів

Операція агрегації даних

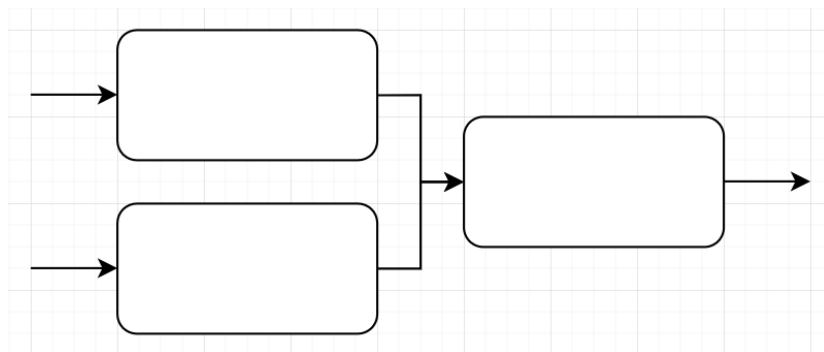


Рис 3.5. Операція агрегації даних

Операція агрегації даних - E_{agr} , це операція, що описує процес передачі вихідних даних багатьох функціональних сервісів на вхід одного функціонального сервісу.

Тоді математична модель даної операції буде виглядати так

$$S_f[E_{agr}(\{ S_{fi} \}, S_{f1})] = \{ M_{bx} = \{ S_{fi}M_{bx} \}, M_{вих} = S_{f1}M_{вих} \}, \text{ де:}$$

M_{bx} – вхідні дані блоку, що буде отримано

$M_{вих}$ – вихідні дані блоку, що буде отримано

$S_{fi}M_{bx}$ – вхідні дані кожного функціонального сервісу

$S_{fi}M_{вих}$ – вихідні дані останнього функціонального сервісу

Операція логічного вибору

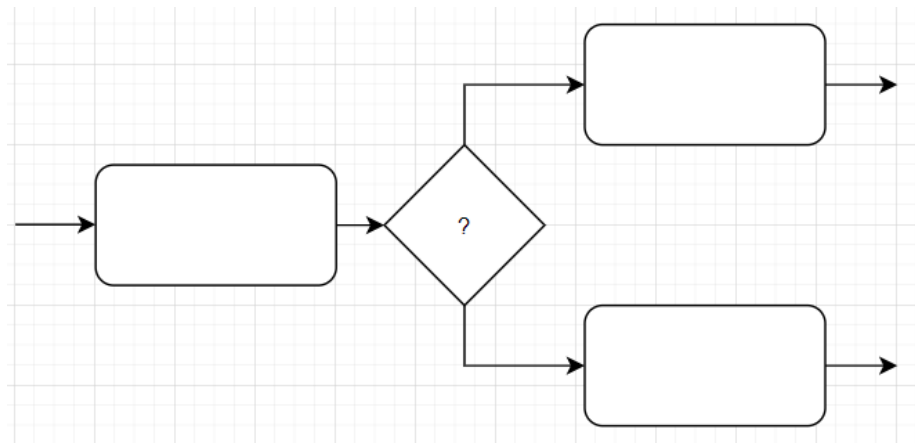


Рис 3.6. Операція логічного вибору

Операція агрегації даних – E_{sel} , це операція, що описує процес передачі вихідних даних одного функціонального сервісу на вхід одного з інших функціональних сервісів.

Тоді математична модель даної операції буде виглядати так

$$S_f[E_{sel}(S_{f1}, \{ S_{fi} \})] = \{ M_{bx} = S_{f1}M_{bx}, M_{вих} = \{ S_{fi}M_{вих} | C_i \} \}, \text{ де:}$$

M_{bx} – вхідні дані блоку, що буде отримано

$M_{вих}$ – вихідні дані блоку, що буде отримано

$S_{fi}M_{bx}$ – вхідні дані першого функціонального сервісу

$S_{fi}M_{вих}$ – вихідні дані другого функціонального сервісу

C_i – умова для передачі даних в i -тий функціональний сервіс

Тоді процес побудови робочого процесу можна представити у вигляді набору функціональних блоків, наприклад:

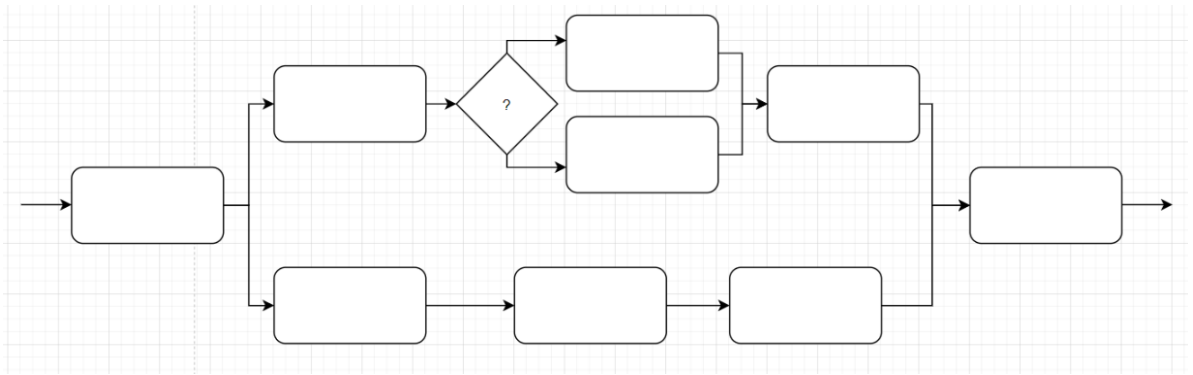


Рис 3.7. Схематичний приклад робочого процесу

При натисканні користувачем кнопки «Зберегти робочий процес» дані будуть передані до системи, буде проаналізовано вибрані сервіси та за можливості запропоновано оптимізація деяких частин робочого процесу.

Після підтвердження користувачем збереження робочого процесу, останній буде додано до бази знань та буде доступний для виконання заданій групі користувачів.

Над усіма побудованими робочими процесами можуть бути виконані такі дії, як:

- Перегляд зформованого бізнес-процесу в веб інтерфейсі
- Редагування даного бізнес-процесу
- Внесення вхідних даних та виконання процесу

Примітка: редагування бізнес-процесу дозволено лише користувачу, що створив даний бізнес процес та спеціалісту предметної області.

3.8. Режим відлагодження

В режимі відлагодження робочого процесу користувачу надається можливість переглядати, редагувати та тимчасово зберігати дані, що проходять через робочий процес, в системі для перевірки правильності виконання робочого процесу.

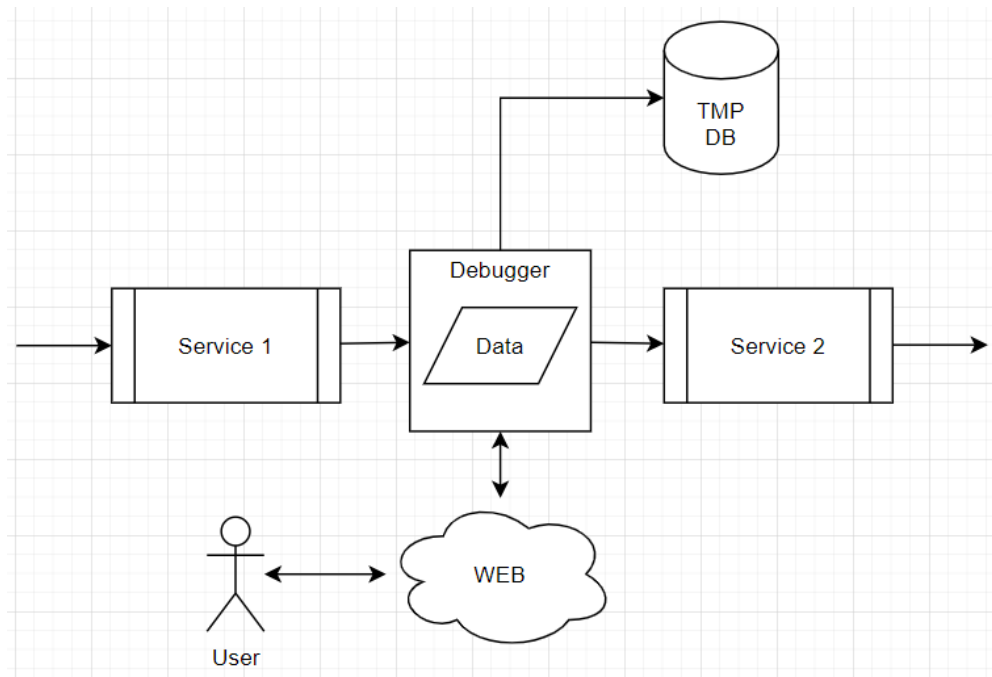


Рис 3.7. Схема режиму відлагодження

Висновки

Запропоновано формалізований опис робочих процесів, який дозволив врахувати обмеження робочої області та налаштувати процес автоматичного виконання та відслідковування стану цих робочих процесів.

Розроблено формалізований опис предметної області, що дозволить врахувати обмеження предметної області під час включення сервісу до складу робочого процесу за рахунок введення логічних правил – зв'язків між функціональними сервісами, предметною областю та робочими процесами.

Запропоновано метод автоматизованої побудови робочих процесів, який може бути швидко побудовано за рахунок врахування обмежень предметної області.

Запропонований метод дозволяє автоматично редагувати робочі процеси та створювати нові на основі заданих вхідних даних, що зменшить витрату часу при виконанні цих дій.

РОЗДІЛ 4.

Дану модель було проаналізовано та представлено у вигляді блок-схеми:

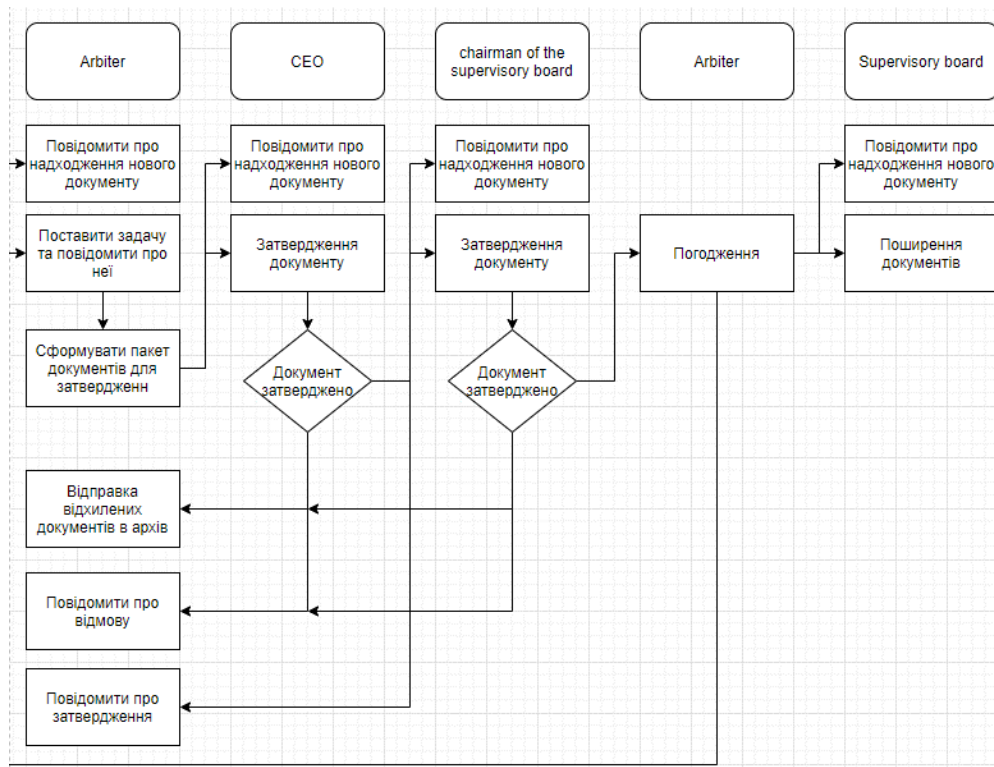


Рис 4.2. Блок-схема робочого процесу, частина 1

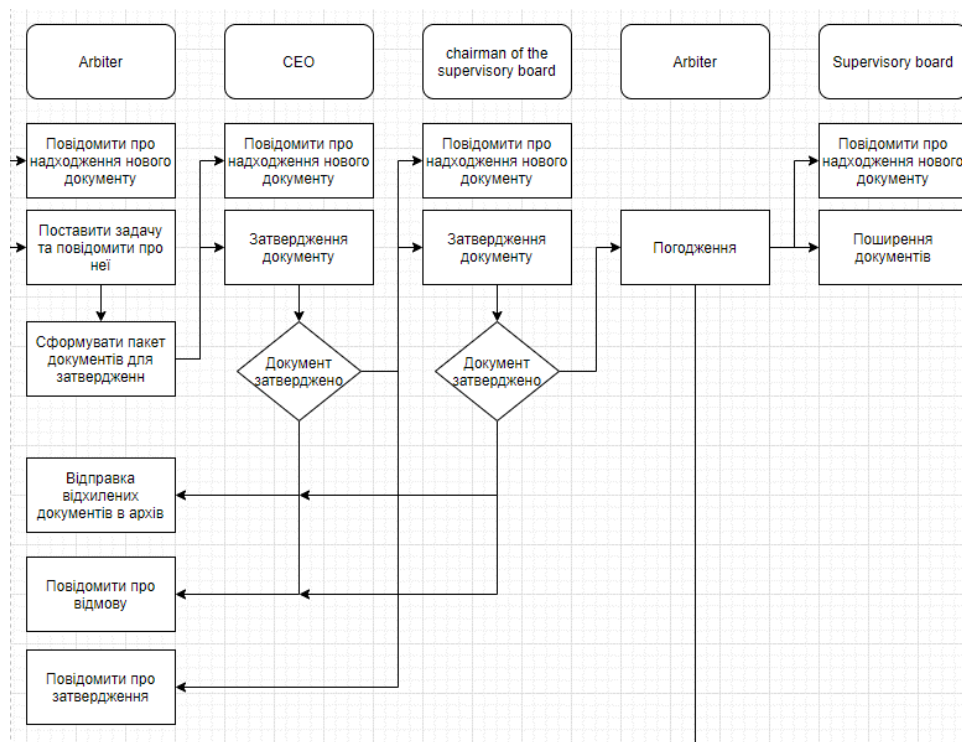


Рис 4.3. Блок-схема робочого процесу, частина 2

4.1. Модель даних

Сервіс може бути одного з таких типів:

- Executor – виконавець, функціональний сервіс;
- Selector – виборець, сервіс вибору наступного сервісу;
- Collector – збирач, сервіс об'єднання даних одного типу за областю допустимих значень
- Informator – інформатор, сервіс, що відправляє повідомлення користувачу;
- Tool – утиліта, системний сервіс для взаємодій різних видів;

4.1.1. Функціональний сервіс

Функціональний сервіс (S_{executor}) – сервіс, що приймає на вхід набір даних, виконує операцію над ними та віддає на вихід нові дані. Збереження функціональних сервісів виконується за допомогою файлової системи, а мета-описи сервісів знаходяться в базі даних. Всі мета-описи функціональних сервісів кожної предметної області зберігаються в окремих колекціях бази даних.

Для вибраного робочого процесу використовуються такі функціональні сервіси:

- DocumentChecker – сервіс перевірки документів користувачами
- FormDocumentPack – сервіс формування пакету документів для поширення в системі

Для сервісу перевірки документів вибрана така модель даних

$I_{\text{executor}} = \{ I_{\text{from}}, I_{\text{checkBy}}, I_{\text{doc}}, I_{\text{to}} \}$, де:

$I_{\text{from}} = \{ N, D, T \}$

N – from

D – ініціатор перевірки документу

T – User

$I_{\text{checkBy}} = \{ N, D, T \}$

N – checkBy

D – особа, яка повинна перевірити документ

T – User

$I_{\text{doc}} = \{ N, D, T \}$

N – document

D – документ для перевірки

T – Document

$I_{\text{to}} = \{ N, D, T \}$

N – to

D – сервіс, в який буде передано результат перевірки

T – Service (S_s)

$O = \{ O_{\text{status}}, O_{\text{notes}} \}$, де:

$O_{\text{status}} = \{ N, D, T \}$

N – status

D – статус перевірки документа

$T = \text{enum } \{ \text{Approved} \mid \text{Edit} \mid \text{Rejected} \}$

$O_{\text{notes}} = \{ N, D, T \}$

N – notes

D – замітки по перевірці документу

T – string

Для сервісу формування пакету документів вибрана така модель даних:

$I_{\text{FormDocumentPack}} = \{ I_{\text{from}}, I_{\text{signedBy}}, I_{\text{doc}}, I_{\text{to}} \}$

$I_{\text{signedBy}} = \{ N, D, T \}$

N – signedBy

D – особа, яка підписала документ

T – User

4.1.2. Сервіс вибору наступного сервісу

Сервіс вибору наступного сервісу (S_{selector}) – мікросервіс, що приймає на вхід один з завчасно заданих статусів, дані, що повинні бути передані в інший сервіс та список сервісів для кожного статусу.

В запропонованому робочому процесі використано такі сервіси селектори:

- CheckedSelector – сервіс обробки стану перевірки документу
- ApprovedSelector – сервіс обробки стану затвердження документу

Для сервісу вибору сервісу за станом перевірки документу використана модель

$$I_{\text{CheckedSelector}} = \{ I_{\text{status}}, I_{\text{data}}, \{ \text{StatusHandler}_1, \dots, \text{StatusHandler}_n \} \}, \text{ де:}$$

$$I_{\text{status}} - \text{enum} \{ \text{Approved} | \text{Edit} | \text{Rejected} \}$$

$$I_{\text{data}} = \{ \text{Document}, \text{Notes} \}$$

$$\text{StatusHandler}_{\text{onApprove}} - \text{Service}$$

$$\text{StatusHandler}_{\text{onEdit}} - \text{Service}$$

$$\text{StatusHandler}_{\text{onReject}} - \text{Service}$$

Для сервісу вибору сервісу за станом затвердження документу використана модель

$$I_{\text{ApprovedSelector}} = \{ I_{\text{status}}, I_{\text{data}}, \{ \text{StatusHandler}_1, \dots, \text{StatusHandler}_n \} \}, \text{ де:}$$

$$I_{\text{status}} - \text{enum} \{ \text{Approved} | \text{Edit} | \text{Rejected} \}$$

$$I_{\text{data}} = \{ \text{Document}, \text{Notes} \}$$

$$\text{StatusHandler}_{\text{onApprove}} - \text{Service}$$

$$\text{StatusHandler}_{\text{onEdit}} - \text{Service}$$

$$\text{StatusHandler}_{\text{onReject}} - \text{Service}$$

4.1.3. Сервіс інформатор

Сервіс інформатор (S_{informer}) – сервіс, який може проінформувати користувачів системи через вибраний канал зв'язку (email, sms, web-push)

В запропонованому робочому процесі використано такі сервіси інформатори:

- MessageSender – сервіс інформування користувача через канал електронних повідомлень
- ShareDocuments – сервіс поширення затверджених документів користувачам через систему web-push

Для сервісу інформування користувача через канал електронних повідомлень використана така модель даних:

$I_{\text{MessageSender}} = \{ I_{\text{email}}, I_{\text{topic}}, I_{\text{message}} \}$, де:

I_{email} – електронна пошта користувача для повідомлення

I_{topic} – тема повідомлення

I_{message} – текст повідомлення

Для сервісу поширення затверджених документів вибрана така модель даних:

$I_{\text{ShareDocuments}} = \{ I_{\text{document}}, \{ I_{\text{target1}}, \dots, I_{\text{targetn}} \} \}$, де:

I_{document} – затверджений документ для поширення

I_{targetn} – цільовий користувач для поширення інформації

4.1.4. Сервіси взаємодії з користувачем

Сервіс утиліта (S_{tool}) – системний сервіс для різних типів взаємодії користувача та системи.

Було визначено такі сервіси-утиліти:

- WorkflowBuilder – побудовник робочого процесу

- WorkflowExecutor – виконавець робочого процесу

Та такі сервіси-інтерфейси:

- Веб-сервіс для авторизації користувача системи
- Веб-інтерфейс для взаємодії користувача з компонентами робочих процесів у вибраній предметній області
- Веб-інтерфейс для спеціаліста, за допомогою якого можуть бути побудовані робочі процеси в зрозумілому для людини вигляді

Крім сервісів у всіх робочих процесах існують ролі, права та користувачі.

4.2. Об'єкти робочого процесу

В системі існують такі ролі:

- Administrator – адміністратор системи
- Specialist – спеціаліст в певній предметній області
- User – користувач системи

Для вибраного бізнес процесу було використано такі додаткові ролі:

- User – користувач, що може подавати запити на підпис документів
- Checker – користувач, що перевіряє документи
- Arbiter – користувач, що погоджує документи
- Approver – користувач, що затверджує документи
- CEO – директор
- Supervisor – користувач, що може поширювати затверджені документи

В кожній ролі є права. Системні права задані таким списком:

- Read – користувач системи має право переглядати сервіси та процеси

- Edit – користувач системи має право на редагування метаданих сервісів та робочих процесів
- CreateWorkflow – користувач має право на створення нових бізнес процесів
- ChangeRoles – користувач має право змінювати ролі інших користувачів

Додатково в системі використано такі ролі:

- Check – користувач з даним дозволом має можливість перевіряти документи
- View – користувач має можливість переглядати документи
- Edit – користувач має можливість редагувати документи
- Approve – користувач має можливість затверджувати документи
- Sign – користувач має право затверджувати та підписувати документи
- Share – користувач має право на поширення документів

Висновки

Для зберігання мета-описів функціональних сервісів, отримання цих даних, та подальшої обробки для автоматизованої генерації онтологічної моделі було використано серверні компоненти.

Протестовано систему на реальному прикладі робочого процесу та сформовано приклад опису бізнес-процесу та його даних. Описано різні типи сервісів, що дозволяє гнучко керувати потоком даних.

Запропонований підхід протестовано на прикладі алгоритму робочого процесу підпису документу, який показав зменшення часу на формування бізнес процесу та дозволив більш зрозуміло візуалізувати отриманий робочий процес для кінцевого користувача.

РОЗДІЛ 5.

СТАРТАП-ПРОЕКТ

Проведенню маркетинговий аналіз стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

5.1. Опис ідеї проекту (товару, послуги, технології)

Продуктом є система автоматизованої побудови робочих процесів на базі онтології. Детальний зміст наведено в таблиці 5.1.

Таблиця 5.1

Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Користь для користувача</i>
Ідея полягає в підвищенні ефективності побудови робочих процесів за рахунок автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків між функціональними сервісами на основі онтологічної моделі, яку легко корегувати, а також запропонованих інструментальних засобів побудови та модифікації бізнес процесів.	1. Навчальні заклади	Система розповсюдження повідомлень користувачам від викладачів
	2. Автоматизація складних розрахунків на основі заданих математичних правил	Постійно повторювані розрахунки за однаковими схемами будуть займати набагато менше часу у користувача

Продуктом є система побудови робочих процесів, тому далі буде проведено порівняльний аналіз аналогічних проектів. Порівняння проводиться із наступними проектами-конкурентами:

- Microsoft SharePoint;
- UML;

Визначення сильних, слабких та нейтральних характеристик ідеї проекту, порівнюючи із проектами конкурентів наведено в таблиці 5.2.

Таблиця 5.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ n/n	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Microsoft SharePoint	UML			
1.	Вартість обслуговування	Низька	Висока.	Низька			так
2.	Вартість експлуатації	Низька	Середня	Низька		так	
3.	Оціночна швидкість роботи	Висока	Середня	Середня			так
4.	Вірогідність втрати даних	Низька	Низька	Середня		так	
5.	Безвідмовність	Висока	Низька	Висока			так
6.	Безпечність використання	Висока	Середня	Середня			так
7.	Зручність взаємодії із проміжними даними	Висока	Середня	Середня			так
8.	Зручність використання	Висока	Середня	Середня			так

5.2. Технологічний аудит ідеї проекту

В таблиці 5.3. проводиться аналіз технологій на основі, яких створюється продукт запропонованої інформаційної системи.

Таблиця 5.3.

Технологічна здійсненність ідеї проекту

№ n/n	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Аналіз онтології предметної області	Онтологічні моделі збережені в загально-визначеному форматі OWL	Технологія наявна та доступна для використання	Технологія є доступною та стандарт відкритий
2.	Система зберігання даних	Використано нереляційну БД, Як приклад - безкоштовна СКБД MongoDB	Технологія наявна та безкоштовна для використання	Технологія є доступною
3.	Побудова робочих процесів	Власна реалізація системи дозволяє гнучко керувати системою	Технології буде представлено у вигляді проекту з відкритим кодом після патентування	Технологія доступна розробникам та альфа-користувачам

5.3. Аналіз ринкових можливостей запуску стартап-проекту

5.3.1. Аналіз попиту

Аналіз приведений в таблиці 5.4.

Таблиця 5.4.

Попередня характеристика потенційного ринку стартап-проекту

<i>№ n/n</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1	Загальний обсяг продаж, грн/ум.од	Дані відсутні
2	Динаміка ринку (якісна оцінка)	Зростає
3	Наявність обмежень для входу (вказати характер обмежень)	Технологічні
4	Специфічні вимоги до стандартизації та сертифікації	Відсутні
5	Середня норма рентабельності в галузі (або по ринку), %	Відсутні

Оцінивши дані, що наведені в таблиці можна зробити висновок, що ринок є рентабельним та відносно простим для входу, тому що відсутні спеціальні регуляції та відсутність монополістів ринку.

5.3.2. Аналіз пропозиції

В таблиці 5.5 показано особливості конкурентного середовища

Таблиця 5.5.

Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Тип конкуренції є чистим, поки не сформувався ринок і не утворились олігополії	Відсутність домінуючих гравців на ринку, або якщо такі є, то занадто перевантажені	Швидкий розвиток та реалізація додаткових можливостей в проекті
2. Рівень конкурентної боротьби є міжнаціональним	Поширення проекту є доволі простим, так як реалізація може бути розгорнута локально.	Швидке розповсюдження на закордонні ринки збуту та в корпораціях.

Продовження таблиці 5.5.

3. Конкуренція є галузевою	Вирішення даної проблеми може бути виконано лише в інформаційній сфері	Обов'язкова реакція на відгуки користувачів для покращення якості продукту
4. Конкуренція за видами товарів є товарно-видовою.	Проявляється в бажанні створити безкоштовний та зручний для користувача продукт	Збільшення швидкості розробки.
5. Характером конкурентних переваг є неціновий	Використання технології, що набувають популярності та можуть бути цікаві коистувачу.	Акцентована увага на пришвидшенні побудови робочих процесів для зменшення затраченого часу користувачів.
6. За інтенсивності конкуренція є немарочною	Кожен продукт використовує різні технології вирішення проблеми.	Забезпечення конкерентної переваги за рахунок пришвидшеного процесу розробки та тестування.

5.3.3. Аналіз умов конкуренції в галузі

В таблиці 5.6 наведений дедальний аналіз конкуренції у галузі.

Таблиця 5.6.

Аналіз конкуренції в галузі за М. Портером

<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
<i>Навести перелік прямих конкурентів</i>	<i>Визначити бар'єри входження в ринок</i>	<i>Визначити фактори сили постачальників</i>	<i>Визначити фактори сили споживачів</i>	<i>Фактори загроз з боку замінників</i>
На даний момент на ринку є досить великі гравці, хоча вони мають реалізацію, що не задовольняє більшість користувачів.	Можливість входу на ринок є, за допомогою продукту, що буде зручним у використанні та швидким при виконанні, а також потребуватиме	Проект базується лише на власних розробках та розробках із відкритим кодом.	Користувачі не ставлять умови для розробки, тому що проектів у заданій сфері небагато.	

	небагато ресурсів для виконання			
--	------------------------------------	--	--	--

На ринку рівень конкуренції не достатньо високий, тому вихід на ринок не є складним на даний момент. Але ринок розвивається достатньо швидко, тому в найближчий час можуть з'явитись досить сильні гравці на ринку, що може ускладнити подальший розвиток стартапу.

Для забезпечення конкурентноспроможності проекту необхідно забезпечити швидкість розробки, скоротити час виконання операцій в системі та розробити систему відгуків клієнтів для реалізації бажань клієнтів та швидкого усунення програмних недоліків.

5.3.4. Фактори конкурентноспроможності показано в таблиці 5.7.

Таблиця 5.7.

Обґрунтування факторів конкурентноспроможності

<i>№ n/n</i>	<i>Фактор конкурентноспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
1.	Вартість експлуатації	Проект безкоштовний, користувачі можуть розгорнути його на своїх серверах або на хмарних сервісах, що дозволить мінімізувати витрати на обслуговування
2.	Безпечність використання	Завдяки тому, що проект розгорнуто локально, всі дані користувачів зберігаються виключно локально
3.	Зручність взаємодії із проміжними даними	В системі наявне середовище відлагодження процесів, що дозволяє зберігати та редагувати проміжні результати у зручному для користувача вигляді
4.	Зручність використання	Взаємодія з системою відбувається за допомогою веб-інтерфейсу, що дозволяє користуватись нею на будь-якому сучасному пристрої

5.3.5. Аналіз сильних та слабких сторін стартап-проекту

Таблиця 5.8.

Порівняльний аналіз сильних та слабких сторін проекту

№ n/n	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із власним продуктом						
			-3	-2	-1	0	+1	+2	+3
1.	Вартість обслуговування	19		+					
2.	Вартість експлуатації	19		+					
3.	Оціночна швидкість роботи	15				+			
4.	Вірогідність втрати даних	15				+			
5.	Безвідмовність	18			+				
6.	Безпечність використання	18			+				
7.	Зручність взаємодії із проміжними даними	19		+					
8.	Зручність використання	19			+				

5.3.6. SWOT-аналіз

Таблиця 5.9.

SWOT- аналіз стартап-проекту

Сильні сторони: 1) дешевизна експлуатації 2) гнучкість системи, 3) автоматизована побудова процесів.	Слабкі сторони: 1) необхідні онтології предметних областей, 2) потрібен спеціаліст предметної області
Можливості: 1) швидке створення робочих процесів 2) невеликі системні вимоги 3) можливість розгорнути локально.	Загрози: 1) зменшення попиту на продукт, 2) поява конкурентів на ринку

Висновки

Проаналізовані можливості розвитку стартап-проекту показують, що вийти на ринок на даний момент є можливим, хоча в найближчий час можуть з'явитись нові конкуренти, тому систему треба розвивати стрімко.

Перевагами розробленого проекту є безкоштовність використання та оплата лише за використання локальних ресурсів або хмарних технологій.

Крім того, дана реалізація автоматизованої побудови робочих процесів є невимогливою до ресурсів сервера, що дозволить зекономити витрати на обладнання.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В роботі запропоновано автоматизовану побудову робочих процесів на базі онтологічних моделей предметної області, функціональних сервісів та робочих процесів, а також операції автоматизованого формування робочого процесу за допомогою зв'язків, що встановлюються між онтологічними моделями.

Проведено огляд та аналіз існуючих методів побудови робочих процесів, який дозволив зробити висновок, робочі процеси, які є основою різноманітних сервісів, будують в недостатньо автоматизованому режимі через відсутність моделей та математичного апарату, який дозволив би легко виконувати певні операції взаємодії елементів таких моделей.

Проведено аналіз інструментальних засобів, що дозволяють зберігати формалізовані описи в довгостроковому сховищі даних, що показав зручність використання нереляційних БД для задач даного типу.

Проаналізовано методи побудови робочих процесів робочих процесів на базі онтологічних моделей, які є недостатньо автоматизованими або зовсім неавтоматизовані.

Запропоновано метод формування робочих процесів на основі вхідних даних та за правилами відповідності параметрів, а також врахування областей допустимих значень та належності сервісів до одної предметної області, що дозволило підвищити ефективність побудови робочих процесів за рахунок автоматизації, зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків

Використовуючи запропонований підхід щодо побудови робочого процесу можна пришвидшити побудову бізнес-процесів, що дозволить мінімізувати витрати часу користувача за рахунок зменшення обсягу ручної праці та за рахунок використання визначеного набору правил побудови зв'язків між функціональними сервісами на основі онтологічної моделі, яку легко корегувати,

а також запропонованих інструментальних засобів побудови та модифікації бізнес процесів.

Запропонований підхід дозволяє здійснювати швидкий пошук по репозиторію сервісів та бізнес-процесів за рахунок введення та використання ключових слів та прив'язаності сервіса до предметної області.

Проаналізовано поточні реалізації бізнес-процесів та порівняно різні сторони їх реалізацій та реалізації стартап-проекту для розуміння можливостей та перешкод при виході з проектом на ринок та представлення проекту загальному колу користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бакай А: Побудова онтологічних моделей FaaS-сервісів
2. Четверта індустріальна революція [Електронний ресурс] // Режим доступу: <https://www.ukrinform.ru/rubric-society/2385951-ukraina-perehodit-na-cifrovuu-ekonomiku-cto-eto-oznaczaet.html>.
3. Кот. Т: Метод проектування потоків завдань в системі OSS/BSS, 2011, 159с.
4. Kornel Terplan. OSS Essentials: Support System Solutions for Service Providers / Kornel Terplan. - New York : John Wiley, 2001. – 610 pages. .
5. Business Process Model and Notation [Електронний ресурс] // Режим доступу: <http://www.omg.org/spec/BPMN/2.0/>.
6. <http://www.internet-of-services.com/index.php?id=264>.
7. erwin Business Process [Електронний ресурс] // Режим доступу: <https://erwin.com/products/erwin-business-process/>.
8. Sparx Systems [Електронний ресурс] // Режим доступу: <http://www.sparxsystems.com/products/index.html>.
9. Visio [Електронний ресурс] // Режим доступу: <http://office.microsoft.com/en-us/visio/>.
10. Workflow Patterns [Електронний ресурс] // Режим доступу: <http://www.workflowpatterns.com/>.
11. Workflow Automation [Електронний ресурс] // Режим доступу: https://web.archive.org/web/20130907014418/http://nocsmart.com/index.php?option=com_content&view=article&id=17&Itemid=135.
12. Інформаційна технологія створення та підтримки порталів інженерних знань : автореф. дис. ... канд. техн. наук : 05.13.06 / Р. Л. Новогрудська; Нац. техн. ун-т України "Київ. політехн. ін-т". - Київ, 2015. - 20 с. - укр..
13. Advanced Approach to Web Service Composition.

14. Грабустс П.С. КОНЦЕПЦИЯ ОНТОЛОГИИ ОСНОВАННАЯ НА ЗНАНИЯХ ДЛЯ КЛАСТЕРИЗАЦИИ ЧИСЛОВЫХ ДАННЫХ, Open Semantic Technologies for Intelligent Systems.
15. Egon Börger. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL, Springer, September 2011. – 14 pages..
16. Transformation of BPMN models for Behaviour Analysis: http://www.win.tue.nl/~jmw/_media/public/transformationforbehaviouranalysis.pdf .
17. Baresi, L.; Pezzè, M.: On Formalizing UML with High-Level Petri Nets. In: Agha, G.A.; De Cindio, F.; Rozenberg, G.: Lecture Notes in Computer Science, Vol. 2001: Concurrent Object-Oriented Programming and Petri Nets, Advances in Petri Nets, pages 276-304.
18. Nüttgens, M.; Feld, T.; Zimmermann, V. Business Process Modeling with EPC and UML: Transformation or Integration? // in: Schader, M.; Korthaus, A. (Hrsg.): The Unified Modeling Language - Technical Aspects and Applications, Proceedings (Mannheim, Oktober .
19. Ouyang, C., Verbeek, E., Aalst van der, W.M.P., Breutel, S., Dumas, M., Hofstede ter, A.H.: Formal Semantics and Analysis of Control Flow in WS-BPEL. Technical report (revised version), Queensland University of Technology (October 2005)..
20. Abdukalykov, R. CSE Dept., Concordia Univ., Montreal, QC, Canada Hussain, I. ; Kassab, M. ; Ormandjieva, O.” Quantifying the Impact of Different Non-functional Requirements and Problem Domains on Software Effort Estimation”..
21. Ian J. Taylor. Workflows for e-Science, Scientific Workflows for Grids / Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, Matthew Shields. - Springer; 1st Edition, 2006.- 552 pages..

22. W. van der Aalst. BPM and Workflow Analysis. BPTrends, 5(4): 1-2, April 2007..
23. Kot, T., Reverchuk, A., Globa, L., Schill, A. (2012): A novel approach to increase efficiency of OSS/BSS workflow planning and design. BIS 2012 Conference. Springer, May 2012..
24. National Institute of Standards and Technology. "Software Errors Cost U.S. Economy \$59.5 Billion Annually" (NIST 2002-10). <http://www.nist.gov/public_affairs/releases/n02-10.htm> (2002).
25. What is an Ontology? [Электронный ресурс] // Режим доступа: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
26. Negru, Stefan; Lohmann, Steffen; Haag, Florian (7 April 2014) "VOWL: Visual Notation for OWL Ontologies" [Электронный ресурс]. Режим доступа: <http://vowl.visualdataweb.org/v2/>.
27. Michael L. Pinedo (7 January 2012) "Scheduling: Theory, Algorithms, and Systems": <https://books.google.com/books?id=QRiDnuXSnVwC>.
28. Introducing the Knowledge Graph [Электронный ресурс] // Режим доступа: <https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>.
29. Patel-Schneider, Peter F.; Horrocks, Ian; Patrick J., Hayes (2004-02-10). "OWL Web Ontology Language Semantics and Abstract Syntax". <https://www.w3.org/TR/2004/REC-owl-semantics-20040210/syntax.html>.
30. OWL 2 Web Ontology Language Document Overview (Second Edition) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/owl2-overview/>.
31. RDF Schema 1.1 [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/rdf-schema/>.

32. WebStotm - The smartest JavaScript IDE (integrated development environment) [Электронный ресурс] // Режим доступа: <https://www.jetbrains.com/webstorm/>.
33. Node.js® - As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications [Электронный ресурс] // Режим доступа: <https://nodejs.org/>.
34. What is MongoDB? [Электронный ресурс] // Режим доступа: <https://www.mongodb.com/what-is-mongodb>.
35. Protégé - Free, open-source ontology editor and framework for building intelligent systems [Электронный ресурс] // Режим доступа: <http://protege.stanford.edu>.
36. Musen, Mark (2015). "The Protégé Project: A Look Back and a Look Forward". AI Matters. 1 (4): 4–12. doi:10.1145/2757001.2757003 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883684/>.
37. Kot T. Applying business process modeling method when Telecommunication services development / Kot T., Globa L., Schill A. // 21-st International Crimean Conference «Crimico'2011». Conference materials. — Sevastopol, Crimea, Ukraine — 2011 — Vol.1 — P.457.
38. Ouyang C. From Business Process Models to Process-Oriented Software Systems / C. Ouyang, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, J. Mendling // ACM Transactions on Software Engineering and Methodology — 2009 — Vol.19 — P. 1-37..
39. Kot, T., Reverchuk, A., Globa, L., Schill, A. Method of non-functional requirements balancing when service development. Proceedings of the International Multi-Conference ACS-AISBIS at Szczecin, Poland, May-June 2012, 9 pp..

40. Kot T., Globa L., Schill A. Applying business process modeling method when telecommunication services development. – Microwave and Telecommunication Technology (CriMiCo), 2011, 21-st International Crimean Conference. Proceedings. – Sevastopol, Ukraine, 20.
41. Ryan K. L. Ko. Business Process Management (BPM) Standards: A Survey / Ryan K. L. Ko, Stephen S. G. Lee, Eng Wah Lee. - Business Process Management Journal, Emerald Group Publishing Limited. Volume 15 Issue 5. (2009). – 48 p..
42. J. Mendling, K.B. Lassen, U. Zdun: "On the Transformation of Control Flow between Block-Oriented and Graph-Oriented Process Modeling Languages." IJBPM. Special Issue on Model-Driven Engineering of Executable Business Process Models., vol. 3, num. 2, pp. .
43. J. Mendling, K. Lassen, U. Zdun: "Transformation Strategies between Block-Oriented and Graph-Oriented Process Modelling Languages ." MKWI 2006, Band 2, XML4BPM Track, GITO-Verlag Berlin, 2006, ISBN 3-936771-62-6, pp. 297-312..
44. http://www.softwareag.com/corporate/products/aris_platform/default.asp.
45. Quan Tran “NFR-Assistant: tool support for achieving quality”, Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium..
46. Axel van Lamsweerde “Requirements Engineering: From System Goals to UML Models to Software Specifications”, John Wiley and Sons , 2009, 650 p..
47. E. Yu, P. Giorgini, N. Maiden, J. Mylopoulos “Social Modeling for Requirements Engineering” Cambridge, MA: MIT Press. 2011..
48. Chung, L., Nixon, B. A., Yu, E. and Mylopoulos, J., Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, Boston, 2000..

49. Chung L., "Representing and Using Non-Functional Requirements: A Process Oriented Approach" Ph.D. Thesis, Dept. of Comp.. Science. University of Toronto, June 1993. Also tech. Rep. DKBS-TR-91-1..
50. Breitman, K. K, Leite J.C.S.P. and Finkelstein Anthony. The World's Stage: A Survey on Requirements Engineering Using a Real-Life Case Study. Journal of the Brazilian Computer Society No 1 Vol. 6 Jul. 1999 pp:13:37.
51. Boehm, Barry and In, Hoh. "Identifying Quality-Requirement Conflicts". IEEE Software, March 1996, pp. 25-35..
52. Lindstrom, D.R. "Five Ways to Destroy a Development Project" IEEE Software, September 1993, pp. 55-58..
53. Oberle, D., Bhatti, N., Brockmans, S., Niemann, M., Janiesch, C. (2009). Countering Service Information Challenges in the Internet of Services, Journal of Business & Information System Engineering, 1(5), Gabler Verlag, 2009, pp. 370-390..
54. Objectiver [Электронный ресурс] // Режим доступа: <http://www.objectiver.com/>.